

⑩ 日本国特許庁 (JP)

⑪ 特許出願公開

⑫ 公開特許公報 (A)

昭58-146969

⑬ Int. Cl.³
G 06 F 15/347

識別記号

庁内整理番号
7056-5B

⑭ 公開 昭和58年(1983)9月1日

発明の数 1
審査請求 未請求

(全 7 頁)

⑮ インデックス限定連続演算ベクトルプロセッサ

川崎市幸区小向東芝町1 東京芝
浦電気株式会社総合研究所内

⑯ 特 願 昭57-28905

⑰ 出 願 人 東京芝浦電気株式会社

⑱ 出 願 昭57(1982)2月26日

川崎市幸区堀川町72番地

⑲ 発 明 者 山崎勇

⑳ 代 理 人 弁理士 則近憲佑 外1名

BEST AVAILABLE COPY

明 細 書

1. 発明の名称 インデックス限定連続演算ベクトル
プロセッサ

2. 特許請求の範囲

データが格納されているデータメモリから、あらかじめ指定された順序で連続して読出されインデックス値のグループを複数組記憶できるインデックス・メモリと、上記演算結果または読出データに対しあらかじめ指定された性質の有無を判定する判定手段と、上記判定の対象である演算結果または読出データのインデックス値を上記判定手段の判定結果に応じてVグループ(性質有)またはWグループ(性質なし)に分けて上記インデックスメモリに記憶させる手段と、上記インデックスメモリから指定されたグループのインデックス値を連続して読出す手段と、前記読出されたインデックス値から指定ベクトルまたはマトリクスの要素のデータメモリ中のアドレスを計算する手段とを有することを特徴とするインデックス限定連続演算ベクトルプロセッサ。

3. 発明の詳細な説明

(発明の技術分野)

本発明は科学技術計算でしばしば必要とされるベクトルやマトリクスの計算を高速に行なうプロセッサに関する。

(発明の技術的背景)

科学技術計算では少し規模が大きくなるとベクトルやマトリクスに対する計算を必要とし、計算機時間がかかる。これらのベクトルやマトリクスの各要素は通常、計算機のメモリ中で、そのインデックスの順に一定のアドレス間隔で格納されることが多い。そこでこのようなデータの集合を“配列”と呼ぶ。ベクトルプロセッサ(またはアレイプロセッサ)はこのような、一定のアドレス間隔で格納されているデータに対して、パイプライン技術を用いて、連続的に高速に同一種類の演算を行なうよう構成されており、これによって、配列に対する計算のスピードを上げようとするものである。すなわち従来のベクトルプロセッサは次のような計算を高速に実行することができる。

(例1)

```
DO 10 I=1, 100  
A(I)=B(I)*C(I)  
10 CONTINUE
```

(例2)

```
DO 20 J=1, 99, 2  
X(I, J)=Y(I, J)+Z(J)  
20 CONTINUE
```

しかし実際の応用では次のような要素毎に計算内容が異なる場合がかなり頻繁に現われる。

(例3)

```
DO 30 I=1, 100  
IF(G(I).GT.0.0) A(I)=A(I)+C(I)  
IF(G(I).LE.0.0) A(I)=A(I)*D(J, I)  
30 CONTINUE
```

この場合には、B(I)の正/負(の零)により演算内容が異なり、かつオペランドが異なっているので、パイプラインがその都度中断し、その結果従来のベクトルプロセッサでは高速に処理できないという問題があった。

(第3ステップ)のように、指定したインデックス集合に属するインデックス値に対してのみオペランドアクセスを連続して行なう機能とを備えることによって、このような場合でもパイプライン処理の効率を失なわないようにしたものである。

(発明の効果)

科学技術計算で何かの臨界面値の両側である量の評価式が異なるような場合が非常に多く、このようなアプリケーションでは従来のベクトルプロセッサは非常に能率の悪いものであったが、本発明のベクトルプロセッサにより、そのような場合でも問題なく高効率で演算を実行できる。

また、本発明のベクトルプロセッサは、ある式を評価すると同時に、あらかじめ指定した性質によりその評価結果を分類し、インデックス集合を発生することができるので、判断分岐命令のような効率の悪いプログラムをほとんど用いずに簡潔演算的にプログラムできる。

またスパースマトリクスの場合、マトリクスとベクトルの積の計算などで零要素との積和をでき

(発明の目的)

本発明は上記事情に鑑みて為されたものであり、その目的は指定されたデータの性質によって演算内容が異なる配列の計算についても高速化したインデックス限定連続演算ベクトルプロセッサを提供することにある。

(発明の概要)

上記欠点を克服するため、例えば上記(例3)を次のように考えなす。

(第1ステップ)

$POSG=\{I|G(I)>0\}$; $NEGG=\{I|G(I)\leq 0\}$

(第2ステップ)

FOR $I \in POSG$, $A(I)=A(I)+C(I)$

(第3ステップ)

FOR $I \in NEGG$, $A(I)=A(I)*D(J, I)$

ここで POSG は $G(I)>0$ であるようなインデックス I の集合、NEGG は $G(I)\leq 0$ であるようなインデックス I の集合である。本発明のベクトルプロセッサは、(第1ステップ)のようなインデックス集合を生成し記憶する機能と、(第2ステップ)、

るだけ行をわなないようにすることによって、計算を高速化することができるが、本発明のベクトルプロセッサでは非零要素のみを選択して演算することが容易にできるので、能率的にスパースマトリクスを取りあつかうことができる(従来は極めて困難であった)。

(発明の実施例)

第1図に本発明の一実施例を示す。図において、10はベクトル演算部、20はオペランドアドレス計算部、30はインデックス計算部、50はこれを制御する制御部である。

ベクトル演算部 10 は、被演算データ、演算結果データ、その他のデータを収容するデータメモリ 11、レジスタ(B) 12、レジスタ(B) 13、レジスタ C 14、ALU 15、レジスタ(A) 16、及びアドレスレジスタ(DA) 17を有し、指定されたデータをなわちベクトル又はマトリクス(以下配列)の演算を実行する。

オペランドアドレス計算部 30 は、演算に必要なオペランドをなわちデータメモリ 11 内におけ

る配列のアドレスを計算するもので、乗数の値を収容するレジスタ(MA)21、レジスタ(MB)22、レジスタ(MC)23、ベースの値を収容するレジスタ(BA)24、レジスタ(BB)25、レジスタ(BC)26、乗算器27、加算器28を有する。

インデックス計算部30は、上述した(第1ステップ)に示すようなインデックス集合を生成し記憶するもので、インデックスメモリ31、レジスタ(V)32、レジスタ(W)33、レジスタ(IB₀)34、レジスタ(IB₁)35、レジスタ(I)36、レジスタ(I')37、加算器38、アドレスレジスタ(IA)39、+1加算器40、終了判定回路41及び正/負判定回路42を有する。このベクトルプロセッサは基本的に次の演算を行なうことができる。

$$\text{FOR } I \in U \begin{cases} A(I) = \text{FUNC}(B(I), C(I)) \\ V = \{I | P(A(I))\} \\ W = \{I | > P(A(I))\} \end{cases}$$

ここに $\text{FUNC}(X, Y)$ は ALU15 のもつ 2 項スカラ演算(加減乗除算など)、 $P()$ は次の 3 種類の

たは I' と乗数 MA, MB, MC とベースの値 BA, BB, BC から、乗算器 27 と加算器 28 を用いて、実際のデータメモリアドレス DA を計算することができる。

演算を行すべき配列の部分集合は、そのインデックス I の集合として、インデックスメモリ 31 に記憶させるがその形式はインデックス・チェインの形式である。ある集合(例えば POSG)とその補集合(NEGG)とは同時に発生させることができ、1つのインデックスベクトルエリアに 2本のチェインとして記憶される(第2図参照)。このインデックスベクトルエリアの先頭番地を IB とすると、インデックス I に対応するインデックスベクトル要素のアドレスは $IB+I$ であり、そこには、この I と同一グループの次の I の値が格納されている。第2図は I の降順にチェインが形成されているが、昇順でもまたランダムでもかまわない。最後のインデックス値の対応箇所には終端記号(4)が書き込まれる。

このような部分集合(インデックスチェイン)

命題関数: ① $X > 0$ の時真、② $X \leq 0$ の時真、③ $X = 0$ の時真、のいずれか。

U, V, W はインデックスメモリ 31 に、 $A(I)$, $B(I)$, $C(I)$ はデータメモリ 11 に格納される。命題関数 $P()$ は判定回路 42 によって判定される。 $A(I)$, $B(I)$, $C(I)$ はベクトル(すなわち 1 次元の配列)の時は、データメモリ 11 のアドレス順に各要素を格納し、従って、その最初の要素($A(0)$ 等)のアドレスを BA とすると、一般に $A(I)$ のアドレスは、 $BA+I$ となる。すなわちベクトルの時は MA, MB, MC は 1 となる。一方 $A(I)$, $B(I)$, $C(I)$ が 2 次元以上の配列の一部、例えば $A(I)$ が 3 次元配列 $AA(X, Y, Z)$ で $X=J$, $Y=K$, $Z=1$ などである時、 $AA(0, 0, 0)$ のアドレスを BA' とすると $A(I) = AA(J, K, 1)$ のアドレスは $BA'+J+X_{\text{MAX}}*K+X_{\text{MAX}}*I$ となるから、 $BA'+J+X_{\text{MAX}}*K$ を改ためて BA とし、 $X_{\text{MAX}}*Y_{\text{MAX}}$ を MA とすると、 $BA+MA*I$ となる。

このようにオペランドアドレス計算部 20 では 3 組の配列 A, B, C に対し、インデックス値 I を

の生成方法(第1ステップ)の一例を説明する。

① V レジスタ 32 及び W レジスタ 33 に終端記号(4)をセットする。I レジスタ 36 に 0 をセットする(又は $I=X$)とする。

② データメモリ 10 からデータ $G(I)$ を Read する。

③ 判定回路 42 により、データ $G(I)$ の判定を行なう。

(i) $G(I) > 0$ ならば、V レジスタ 32 の内容をインデックスメモリ 31 の IB_1+I 番地に書き込み、V レジスタ 32 に I レジスタの内容をセットする。

(ii) $G(I) \leq 0$ ならば、W レジスタ 33 の内容をインデックスメモリ 31 の IB_1+I 番地に書き込み、W レジスタ 33 に I レジスタ 36 の内容をセットする。

④ +1 加算器 40 により I レジスタ 36 の内容を +1 する(またはインデックスメモリ 31 の IB_1+I 番地の内容を I レジスタ 36 にセットする)。

⑤ 終了判定回路 41 により、I レジスタ 36 の内容が I の最大値(または終端記号)であれ

は次の④へ、そうでなければ②へもどる。

④ POSG=V, NEGG=W とする。

すなわち V レジスタ 32 の内容、第 2 図では "9" を POSG とし、W レジスタ 33 の内容、第 2 図では "10" を NEGG とする。①、④、⑤のステップで () 内を用いた時は部分集合の部分集合つまり積集合を求めることができる。(XかつG(I)>0 など)

次に第 2 ステップでは I POSG に対する演算を行なう。この場合、まず POSG すなわち "9" を I レジスタ 36 にセットし、I の更新タイミング(第 3 図の (a)) で、 $I+IB_0$ をアドレスとしてインデックスメモリ 31 を Read し、読出したインデックス値を I レジスタ 36 にセットするように動作する。これにより、 $G(I)>0$ である I の値のみが次々と I レジスタ 36 にセットされ、それに対し同一演算 FUNC が ALU15 で行なわれた。

この動作を第 3 図を用いて説明する。本実施例では I から始めて 5 段のパイプライン制御を行なっている。第 3 図では第 (n) サイクルから第 (n

サイクルの初めで C レジスタ 14 にセットされる。

ALU 15 は FUNC の実行に 3 サイクル(第 (n+3)~(n+5) サイクル)要し、この間 B' レジスタ 13 及び C レジスタ 14 の内容は保持されている。

ALU15 の演算結果はデータ A (5) としてデータメモリ 11 に収容する必要上、第 (n+3) サイクルの初めで I の値を更新("5" から "3" へ)する直前に I レジスタ 36 の内容は I' レジスタ 37 にセットされる。この I' レジスタ 37 に再セットされた値 "5" を用いて、第 (n+5) サイクルでオペランドアドレス計算部 20 はデータ A (5) のアドレスを MA レジスタ 21 及び BA レジスタ 24 の内容を用いて算出し、第 (n+5) サイクルの初めで DA レジスタ 17 にセットする。しかして、第 (n+6) サイクルの初めで ALU の演算結果 (FUNC (13(5), C(5))) が A レジスタ 16 にセットされると、この内容はデータ A (5) としてデータメモリ 11 に Write される。

このように、ALU 15 による FUNC の実行が行な

われる。第 (n) サイクルの初めで I レジスタ 36 には "5" がセットされると、オペランドアドレス計算部 20 はデータ B (5) のアドレスを計算する。すなわち、乗算器 27 は I レジスタ 36 の値 5 と MB レジスタ 22 の乗数を乗算し、この乗算結果に BB レジスタ 25 のベースを加える。この結果得られた値 $BB+5\cdot MB$ は第 (n+1) サイクルの初めで DA レジスタ 17 にセットされる。次にデータメモリ 11 のアドレス $BB+5\cdot MB$ の内容すなわちデータ B (5) を Read し、第 (n+2) サイクルの初めで B レジスタ 12 にセットされた後、第 (n+3) サイクルの初めで B' レジスタ 13 にセットされる。

一方、同様にして第 (n+1) サイクルで MC レジスタ 23 及び BC レジスタ 26 の内容を用いてオペランドアドレス計算部 20 はデータ C (5) のアドレス $BC+5\cdot MC$ を求め、第 (n+2) サイクルの初めで DA レジスタ 17 にセットする。次に DA レジスタ 17 の内容で指定されるデータメモリ 11 の内容すなわちデータ C (5) が Read され、第 (n+3) サイクルの初めで DA レジスタ 17 にセットされる。次にデータメモリ 11 のアドレス $BB+5\cdot MB$ の内容すなわちデータ B (5) を Read し、第 (n+2) サイクルの初めで B レジスタ 12 にセットされた後、第 (n+3) サイクルの初めで B' レジスタ 13 にセットされる。

われる 3 サイクル間、その第 1 サイクルでは直前の演算結果のデータメモリ 11 への書き込みとオーバーラップし、第 2、第 3 サイクルでは夫々次に演算すべきオペランドの読出しとオーバーラップしている。更にこれらデータメモリ 11 への書き込み又は読出しとオーバーラップして、夫々次のサイクルで必要となるオペランドのアドレスがオペランドアドレス計算部 20 により実行されている。

このようにして I POSG の演算が実行され、I レジスタ 36 に終端マーク (a) がセットされると終了判定回路 41 はこれを検出し、I レジスタ 36 に NEGG すなわち "10" をセットし、以下同様にして I NEGG の演算が実行される。

なお上記実施例ではインデックス値のグループを 2 組に分けたが一般に複数組に分けその夫々について異なる演算を実行してもよい。また、上記実施例ではデータメモリ内の値 $G(I)$ に基づいてインデックスのグループ分けを行っているが、ALU の演算結果を用いることもできる。

4. 図面の簡単な説明

第 1 図は本発明の一実施例を示す図、第 2 図及び第 3 図は本発明の一実施例を説明するための図である。

1.0 ... ベクトル演算器

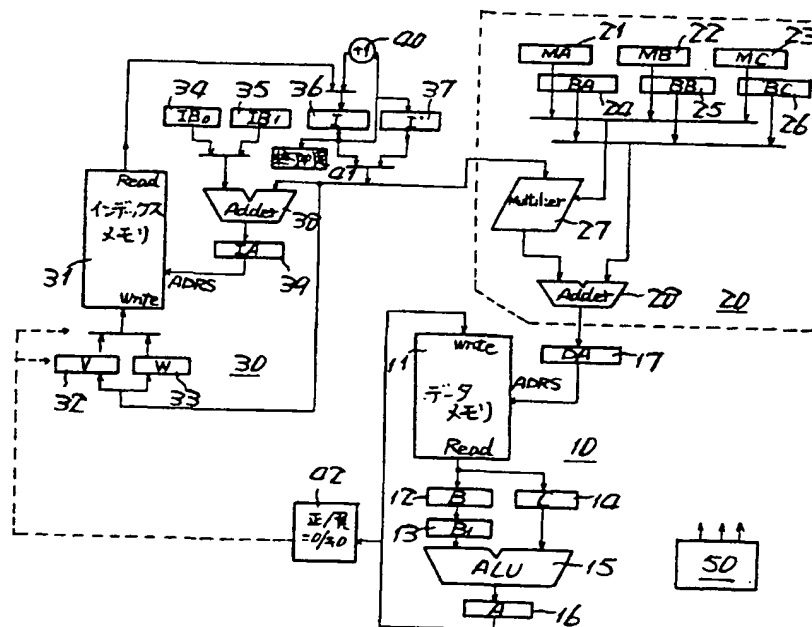
20 … オペランドアドレス計算部

3.0 ... インデックス計算部

5.0 ... 制御部

代理人 弁理士 則 近 藤 佑
(任 務 1 名)

第 1 回

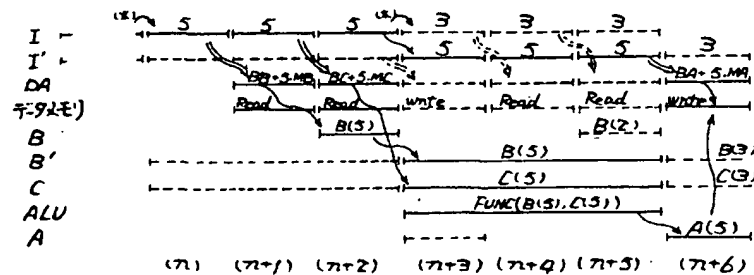


第 2 図

I	G(I)	U/V
0	正	0
1	負	1
2	負	0
3	正	2
4	負	3
5	正	5
6	正	6
7	正	4
8	負	7
9	正	8
10	負	

POSG=4
NEGG=10

第 3 図



手 続 補 正 書 (自 発)

昭和 57 年 9 月 22 日

特許庁長官 殿

1. 事件の表示

昭和 57 年特願第 28905 号

2. 発明の名称

インダクティブ固定連続演算ベクトルプロセッサ

3. 補正をする者

事件との関係 特許出願人

(307) 東京芝浦電気株式会社

4. 代理人

〒100

東京都千代田区内幸町1-1-6

東京芝浦電気株式会社東京事務所内

(7317) 舟橋士 剛 近 藤 佑

5. 補正の対象

- (1) 明細書の特許請求の範囲の欄
- (2) 明細書の発明の詳細な説明の欄
- (3) 図面

6. 補正の内容

- (1) 明細書の特許請求の範囲を別紙の通り補正する。
- (2) 明細書の発明の詳細な説明を以下の正誤表の通り訂正する。

箇所	正	誤
3頁16行	(または零)	(の零)
7頁17行	$W = \{I\} \cap P(A \cap D)$	$W = \{I\} > P(A \cap D)$
8頁13行	$Z = I$	$Z = 1$
8頁16行	$\dots X_{\max} * K + X_{\max} * Y_{\max} * I \dots$	$\dots X_{\max} * K + X_{\max} * I \dots$
10頁5行	データメモリ 11	データメモリ 10
10頁17行	$I B_0 + I$	$I B_1 + I$
11頁9行	$I \in \text{POSG}$	$I \text{ POSG}$
11頁17行	行なわれる。	行なわれた。
13頁14行	始めで	初めて
13頁17行	$(B(5), O(5))$	$(13(5), O(5))$
14頁9行	$I \in \text{POSG}$	$I \text{ POSG}$

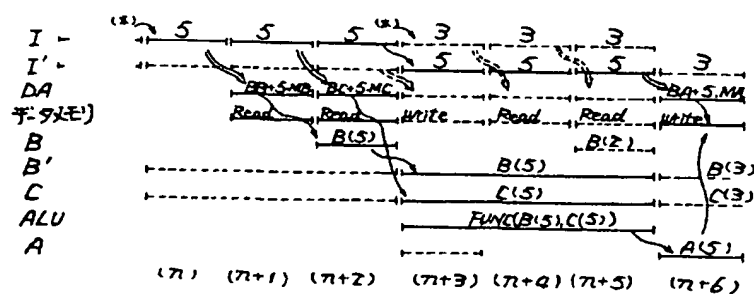
- (3) 図面の内、第3図を別紙の通り補正する。

以 上

計算求の範囲

データが格納されているデータメモリから、あらかじめ指定された順序で連続して読出されるデータに対し、指定された同一種の演算を連続して高速に実行できるベクトルプロセッサにおいて、インデックス値のグループを複数組記憶できるインデックス・メモリと、上記演算結果またはデータメモリからの読出データに対しあらかじめ指定された性質の有無を判定する判定手段と、上記判定の対象である演算結果または読出データのインデックス値を上記判定手段の判定結果に応じてVグループ（性質有）またはWグループ（性質なし）に分けて上記インデックスメモリに記憶させる手段と、上記インデックスメモリから指定されたグループのインデックス値を連続して読出す手段と、前記読出されたインデックス値から指定ベクトルまたはマトリクス要素のデータメモリ中のアドレスを計算する手段とを有することを特徴とするインデックス限定連続演算ベクトルプロセッサ。

第 3 図



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.